Yee & Associates, P.C.

4100 Alpha Road
Suite 1100
Dallas, Texas 75244

Main No. (972) 385-8777
Facsimile (972) 385-7766

Facsimile Cover Sheet

To: Commissioner for Patents for **Examiner Kambiz Zand** **Group Art Unit 2132**	Facsimile No.: **571/273-8300**
From: Kim Gault Legal Assistant to Wayne Bailey	No. of Pages Including Cover Sheet: **37**

Message:

Enclosed herewith:

- Transmittal Document; and
- Appeal Brief.

Re: Application No. 09/865,246
Attorney Docket No: **YOR920010310US1**

Date: Monday, January 30, 2006

Please contact us at (972) 385-8777 if you do not receive all pages indicated above or experience any difficulty in receiving this facsimile.

PLEASE CONFIRM RECEIPT OF THIS TRANSMISSION BY FAXING A CONFIRMATION TO 972-385-7766.

RECEIVED
CENTRAL FAX CENTER

JAN 3 0 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Swimmer et al.**

Serial No.: **09/865,246**

Filed: **May 25, 2001**

For: **Method and Apparatus for Repairing Damage to a Computer System Using a System Rollback Mechanism**

§ Group Art Unit: **2132**
§
§ Examiner: **Zand, Kambiz**
§
§ Attorney Docket No.: **YOR920010310US1**
§

54105
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Certificate of Transmission Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (571) 273-8300 on January 30, 2006.

By: Kim Gault
Kim Gault

TRANSMITTAL DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
ENCLOSED HEREWITH:

- Appeal Brief (37 C.F.R. 41.37)

A fee of $500.00 is required for filing an Appeal Brief. Please charge this fee to Deposit Account No. 50-3533. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to Deposit Account No. 50-3533. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Deposit Account No. 50-3533.

Respectfully submitted,

Duke W. Yee
Registration No. 34,285
YEE & ASSOCIATES, P.C.
P.O. Box 802333
Dallas, Texas 75380
(972) 385-8777
ATTORNEY FOR APPLICANTS

RECEIVED
CENTRAL FAX CENTER

JAN 3 0 2006

Docket No. YOR920010310US1 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Swimmer et al.** §
§ Group Art Unit: **2132**
Serial No. **09/865,246** §
§ Examiner: **Zand, Kambiz**
Filed: **May 25, 2001** §
§
For: **Method and Apparatus for Repairing Damage to a Computer System Using a System Rollback Mechanism** §
§

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Certificate of Transmission Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (571) 273-8300 on January 30, 2006.
By: Kim Gault
Kim Gault

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on November 30, 2005.

The fees required under § 41.20(B)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

02/01/2006 EFLORES 00000028 503533 09865246
01 FC:1402 500.00 DA

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: Lenovo Group Limited.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-67

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: none
2. Claims withdrawn from consideration but not canceled: none
3. Claims pending: 1-67
4. Claims allowed: none
5. Claims rejected: 1-67
6. Claims objected to: none

C. CLAIMS ON APPEAL

The claims on appeal are: 1-67

STATUS OF AMENDMENTS

No amendment after final was filed for this case.

RECEIVED
CENTRAL FAX CENTER

JAN 3 0 2006

SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is generally directed to a technique for repairing damage caused by an unauthorized intrusion of a data processing system.

A. CLAIM 1 - INDEPENDENT

Claim 1 is specifically directed to a method in a data processing system for protecting data from damage. Data is journaled to form journaled data, where journaling the data comprises maintaining a previous state of the data for subsequent, optional restore of the data to the previous state. A determination is made as to whether a virus is present in the data processing system after journaling of the data has begun. Responsive to an identification of a virus, the data is restored using the journaled data, thereby repairing damage caused by an unauthorized intrusion of the data processing system (Specification page 9, line 4 – page 10, line 9 and Figure 3, element 220; Specification page 12, line 17 – page 13 line 10 and Figure 5, all blocks).

B. CLAIM 13 - INDEPENDENT

Claim 13 is specifically directed to a method in a data processing system for repairing damage to data. The state of a data object is saved in response to a request to access the data object by a process. Pattern matching of a set of actions taken within the data processing system is performed. A determination is made as to whether an unauthorized intrusion has occurred in response to performing the pattern matching and if so, a rollback is initiated to return the data object back to its saved state (Specification page 9, line 4 – page 10, line 9 and Figure 3, element 220; Specification page 12, line 17 – page 13 line 10 and Figure 5, all blocks).

C. CLAIM 22 - INDEPENDENT

Claim 22 is specifically directed to an intrusion protection system for use in a data processing system. Such intrusion protection system includes a sensor filter that receives requests to access data within the data processing system from a process. This intrusion protection system also includes a pattern matcher that receives actions initiated by the process,

compares the actions to a pattern to form a comparison, determines whether an unauthorized intrusion has occurred, generates a first indication in response to an identification of an absence of an unauthorized intrusion, and generates a second indication to restore the data to a prior state in response to an identification of the unauthorized intrusion. This intrusion detection system also includes a journaler that (i) journals data in response to accessing of the data and (ii) restores the data to the prior state in response to the indication by the pattern matcher. The data is journaled until the first indication is generated by the pattern matcher (Specification page 9, line 4 – page 10, line 9 and Figure 3, element 220; Specification page 12, line 17 – page 13 line 10 and Figure 5, all blocks).

D. CLAIM 24 – INDEPENDENT

Claim 24 is a system claim corresponding to method Claim 1, and the summary of Claim 1 is applicable for Claim 24, and thus is hereby incorporated by reference.

E. CLAIM 25 – INDEPENDENT

Claim 25 is a system claim corresponding to method Claim 13, and the summary of Claim 13 is applicable for Claim 25, and thus is hereby incorporated by reference.

F. CLAIM 26 – INDEPENDENT AND MEANS-PLUS FUNCTION

Claim 26 is a system claim corresponding to method Claim 1, and the summary of Claim 1 is applicable for Claim 26, and thus is hereby incorporated by reference.

Claim 26 is also a means-plus-function claim, and the structure corresponding to each of the recited means-for elements is described at Specification page 7, line 12 – page 9, line 3 and shown by reference character 200 in Figure 2.

G. CLAIM 38 – INDEPENDENT AND MEANS-PLUS FUNCTION

Claim 38 is a system claim corresponding to method Claim 13, and the summary of Claim 13 is applicable for Claim 38, and thus is hereby incorporated by reference.

Claim 38 is also a means-plus-function claim, and the structure corresponding to each of the recited means-for elements is described at Specification page 7, line 12 – page 9, line 3 and shown by reference character 200 in Figure 2.

H. CLAIM 47 – INDEPENDENT

Claim 47 is a program product claim corresponding to method Claim 1, and the summary of Claim 1 is applicable for Claim 47, and thus is hereby incorporated by reference.

I. CLAIM 59 – INDEPENDENT

Claim 59 is a program product claim corresponding to method Claim 13, and the summary of Claim 13 is applicable for Claim 59, and thus is hereby incorporated by reference.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. GROUND OF REJECTION 1 (Claims 1-67)

Claims 1-67 stand rejected under 35 U.S.C. § 103 as being unpatentable over Conklin et al. (US 5,991,881 A) in view of Cozza (US 5,473,769 A).

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 1-67)

Claims 1-67 stand rejected under 35 U.S.C. § 103 as being unpatentable over Conklin et al. (US 5,991,881 A) in view of Cozza (US 5,473,769 A).

A.1. Claims 1, 3-7, 24, 26, 28-32, 47 and 49-53

With respect to Claim 1, such claim recites the claimed feature of "journaling the data to form journaled data, wherein *journaling the data comprises maintaining a previous state of the data for subsequent, optional restore of the data to the previous state*". As can be seen, the claimed journaling of data is with respect to data that can be restored to a previous state. None of the cited references teach or otherwise suggest such journaling of data, and thus it necessarily follows that none of the cited references teach or otherwise suggest the claimed steps of (1) "determining whether a virus is present in the data processing system *after journaling of the data has begun*" (since there is no teaching of journaling of data), or (2) "responsive to an identification of the virus, *restoring the data using the journaled data*" (since there is no teaching of the claimed journaled data). These claimed features advantageously provide an ability to repair damage caused by an unauthorized intrusion of a data processing system, and accomplishes this advantage by pro-actively journaling data such that it can be restored in the event of such unauthorized intrusion. None of the cited references teach or suggest the claimed features described above, or their resulting advantages, as will now be described in detail.

The cited Conklin reference teaches a system that logs network traffic data *upon detection of an unauthorized intrusion* (a re-active mode of operation, where network traffic is logged in response to detection of an unauthorized intrusion), but does not teach or otherwise suggest *journaling of data prior to detecting a virus* (a pro-active mode of operation, where data is journaled prior to virus determination). Claim 1 explicitly recites 'determining whether a virus is present in the data processing system *after journaling of the data has begun*' (emphasis added). The cited Conklin reference is the sole reference being relied upon as teaching this claimed feature, and thus it is shown that there is at least one claimed feature not taught or suggested by the cited references.

The cited Cozza reference is concerned about the length of time it takes to perform a scan of viruses in a computer system, and is specifically directed to particular virus scanning techniques that reduce the scan time to scan for a virus. This reference does not teach any type of *data restore operation* that uses journaled data. Claim 1 specifically recites "responsive to an identification of the virus, restoring the data using the journaled data". In rejecting Claim 1, the Examiner acknowledges that the cited Conklin reference does not disclose restoring the data using journaled data, but states that the cited Cozza reference discloses restoring the data using the journaled data at col. 2, lines 49-67. Appellants urge that there, Cozza states:

> "The method and apparatus of the present invention for scanning files for computer viruses relies on the fact that viruses invariably change the file or volume they infect. Consequently, information detailing the initial "state" of an uninfected file or volume can be "cached" or securely saved to disk or other non-volatile storage medium. The cached information is dependent not only on the type of machine the scanning program is running on, but also on viruses' method of infection on that type of machine. The stored information can be tailored to meet the variety of situations found in present and future computing environments.
>
> Once the initial "state" information has been stored to a disk or other non-volatile storage medium, the method and apparatus of the present invention can use this cached information in future virus scans to determine what files and/or volumes have changed in a way indicative of most virus infections. In many applications this information alone is enough to eliminate the need to scan a file/volume for most, if not all, viruses."

As can be seen, this passage describes a technique for *scanning files* for a virus. It does not describe in any way subsequent data restore actions that are performed as a result of detecting a virus. As a part of this Cozza volume scan operation, information detailing the initial "state" of an uninfected file or volume is cached or saved to disk such that is can be used when comparing

a current state of a file or volume during a scan of the file/volume *in order to determine what files or volumes have changed.* Thus, this technique is specifically directed to a way of scanning a file/volume to reduce the time it takes for such scan, because in many instances, this information alone is enough to eliminate the need to scan a file/volume for most, if not all, viruses. This cited passage does not describe any action – such as restoring data using journaled data – that results from detection of a virus. Claim 1 expressly recites "responsive to an identification of the virus, restoring the data using the journaled data".

This missing claimed feature is also evidenced by Cozza's Figure 4, block 58, where a check is made as to whether there are any viruses. If yes, the file cache entry is zeroed out and the file scan ends. The process then loops to the next file on the volume to be scanned. Thus, *the only action described by Cozza that is responsive to an identification of a virus is to zero the file cache entry*. This is described at Cozza col. 5, lines 23-26 and col. 4, lines 20-21, where this operation zeroes out the cache such that the file will be completely scanned in the future. Such zeroing of the cache does not teach or otherwise suggest the claimed feature of "responsive to an identification of the virus, *restoring the data using the journaled data*". It is thus urged that a proper prima facie case of obviousness has not been established with respect to Claim 1, as there are claimed features not taught or suggested by the cited references[1]. Accordingly, the burden has not shifted to Appellants to rebut obviousness[2]. It is thus shown that Claim 1 has been improperly rejected under 35 USC 103[3].

In addition to the above reasons of why Claim 1 is not obvious in view of the cited references, Appellants will now show that there are at least three reasons why the Examiner's analysis that was given in rejecting Claim 1 is erroneous.

First, 'journaling of data' is expressly defined in Claim 1 to comprise 'maintaining a previous state of the data for subsequent, optional restore of the data to the previous state'. In rejecting Claim 1, the Examiner states that (1) the cited Conklin reference teaches the claimed

---

[1] To establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974).

[2] In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. Id.

[3] If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned. *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

'journaling of data', but that (2) the cited Conklin reference does not disclose 'maintaining a previous state of the data for subsequent, optional restore of the data to the previous state'. Appellants urge that this reasoning is logically inconsistent with the definition of 'journaling data' that is expressly recited in Claim 1. If the explicitly defined steps that comprise the defined 'jounaling of data' are not disclosed by Conklin, then it necessarily follows that Conklin cannot teach the claimed 'jounaling of data'. This is error.

Second, the Examiner provides circular reasoning in the rejection of Claim 1. The Examiner states that the cited Conklin reference teaches 'wherein *journaling the data* comprises determining whether a virus is present in the data processing system after *journaling of the data* has begun'. This is circular reasoning in that the Examiner states that the term 'journaling of data' comprises performing an action (determining whether a virus is present) after journaling of data has begun. It is not seen how an action (journaling of data) can be defined to be something that occurs *after this same action* (journaling of data) has begun. This further evidences logically flawed reasoning being given by the Examiner in rejecting Claim 1.

Third, the Examiner changes the definition of 'journaling of data' that is expressly recited in Claim 1. Claim 1 states that journaling the data to form journaled data, *wherein journaling the data comprises maintaining a previous state of the data for subsequent, optional restore of the data to the previous state*. In rejecting Claim 1, the Examiner states "wherein journaling the data comprises determining whether a virus is present in the data processing system after journaling of the data has begun". This is an impermissible, and thus erroneous, re-characterization of an explicit term recited in Claim 1, further evidencing that Claim 1 has been erroneously rejected.

Appellants have thus shown numerous and substantial differences between the invention recited in Claim 1 and the teachings of the cited references, and thus urge that Claim 1 has been erroneously rejected.

A.2. Claims 2, 27 and 48

With respect to Claim 2, Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 2 depends upon).

Further with respect to Claim 2, Appellants urge that none of the cited references teach or otherwise suggest the claimed feature of "responsive to an absence of an identification of the virus, discarding the journaled data". In rejecting Claim 2, the Examiner cites Conklin Figures 6 and 7 and associated text as teaching this claimed step. Appellants urge that while these figures show a discard box and describe an associated discard operation being performed if no virus is identified, *the item that is discarded is the data packet that is received from the network and examined for a potential virus* (col. 5, lines 22-25). A teaching of discarding a received data packet from a network, as described by Conklin, does not teach or otherwise suggest *discarding of the journaled data*, as expressly recited in Claim 2. Thus, Claim 2 is further shown to have been improperly rejected, as a proper prima facie showing of obviousness has not been established.

A.3. Claims 8, 33 and 54

With respect to Claim 8, Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 8 depends upon).

Further with respect to Claim 8, Appellants urge that none of the cited references teach or otherwise suggest the claimed feature of "responsive to an identification of the virus, blocking access to the data by a process accessing the data". In rejecting Claim 8, the Examiner cites Conklin's teaching at col. 5, lines 34-38 as disclosing this claimed feature. Appellants urge that this cited passage describes writing triggering packets to a log file for subsequent processing. Such writing of packets to a log file for subsequent processing does not teach or otherwise suggest *blocking access to the data*, as expressly recited in Claim 8. Thus, Claim 8 is further shown to have been improperly rejected, as a proper prima facie showing of obviousness has not been established.

A.4. Claims 9, 34 and 55

With respect to Claim 9, Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 9 depends upon).

Further with respect to Claim 9, Appellants urge that none of the cited references teach or otherwise suggest the claimed feature of "responsive to an identification of the virus, generating

an indication halting a process accessing the data". In rejecting Claim 9, the Examiner cites Conklin's teaching at col. 5, lines 22-44 as teaching this claimed step. Appellants urge that there, Conklin states:

> "If, there is no indication of an actual or potential intrusion, then the examined packet data is discarded. When a packet or accumulation of packets match a predefined intrusion profile the Intrusion Detection function identifies the network traffic as a reportable activity will construct a data structure which contains a date/time stamp indicating the time of detection, the source and destination Internet Protocol (IP) addresses, an assigned message identifying the event detected. This data structure is passed to the Alert Notification function for processing. When a positive identification of a reportable activity occurs, the entire triggering packet(s) may be written to a log file created in the Evidence Logging function. This log file is then used to hold all ensuing packets associated with this reportable activity event by any one of its identifiable characteristics. For example, the log file written is named using the date/time and name of event detected. The source Internet Process (IP) address is sent to the Evidence Logging function as a controlling parameter so that a secondary logging function may be started which will only capture packets to and from the IP address identified as the source of the intrusion or attack."

As can be seen, when a packet matches an intrusion profile, (1) the network traffic is identified as a reportable activity, (2) a data structure is created which contains information pertaining to the packet, which is passed to the Alert Notification function for processing, (3) the triggering packets are written to a log file for subsequent processing, and (4) a secondary logging function is started which only captures certain subsequent packets. *This passage makes no mention of any type of halting operation*, and thus does not teach or otherwise suggest the claimed step of "responsive to an identification of the virus, generating an indication halting a process accessing the data". Thus, Claim 9 is further shown to have been improperly rejected, as a proper prima facie showing of obviousness has not been established.

A.5. Claims 10, 35 and 56

With respect to Claim 10, Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 10 depends upon).

Further with respect to Claim 10, Appellants urge that none of the cited references teach or otherwise suggest the claimed feature of "wherein the journaled data is accessed by a single process and maintained until a determination is made that the single process is eliminated as a virus candidate". In rejecting Claim 10, the Examiner cites Conklin's teaching at figure 6 and associated text as disclosing this claimed feature. Appellants urge that this cited portion of Conklin does not describe any type of process eliminiation technique, and specifically does not describe any method of eliminating as a virus candidate *the process that is accessing journaled data*. For example, Conklin is examining data packets received across a network for viruses, and is not examining or making virus determinations with respect to internal processes themselves, and thus is not making any type of virus determination with respect to a process that is accessing journaled data, as expressly recited in Claim 10. Thus, Claim 10 is further shown to have been improperly rejected, as a proper prima facie showing of obviousness has not been established.

A.6. Claims 11, 36 and 57

With respect to Claim 11, Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 11 depends upon).

Further with respect to Claim 11, none of the cited references teach or suggest the claimed feature of "wherein the *journaled data* is stored in a protected memory accessible only by the method". In rejecting Claim 11, the Examiner cites Conklin's teaching at figure 9 and associated text as disclosing this claimed feature. Appellants urge that since Conklin does not teach or suggest "journaled data" (which is specifically defined in Claim 1, of which Claim 11 depends upon, and is shown above to not be taught by Conklin in the Claim 1 discussion), it necessarily follows that Conklin does not teach storing such (missing) journaled data in a particular place, as per Claim 11. Thus, Claim 11 is further shown to have been improperly rejected, as a proper prima facie showing of obviousness has not been established.

A.7. Claims 12, 37 and 58

With respect to Claim 12, Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 1 (of which Claim 12 depends upon).

Further with respect to Claim 12, none of the cited references teach or suggest the claimed feature of "wherein the *journaled data* is stored in a data structure located in a protected memory inaccessible by a process" for similar reasons to those given above with respect to Claim 11. Thus, Claim 12 is further shown to have been improperly rejected, as a proper prima facie showing of obviousness has not been established.

A.8. Claims 13, 14, 16-21, 25, 38, 39, 41-46, 59, 60 and 62-67

With respect to Claim 13, Appellants urge that none of the cited references teach or suggest the claimed step of "saving a state of a data object *in response to a request to access the data object by a process*", which is in addition to the claimed step of "performing pattern matching of a set of actions taken within the data processing system". Appellants urge that neither of the cited references teach or suggest both (i) pattern matching of a set of actions *and* (ii) *saving a state of a data object in response to a request to access the data object by a process.* The Examiner states that everything recited in Claim 13, except the rollback operation, is described by Conkin at figs. 6-9 and the associated text. Appellants have reviewed these passages extensively, and can find no teaching of the claimed element of "saving a state of a data object *in response to a request to access the data object by a process*". Thus, there is at least one missing claimed step not taught or suggested by the cited references.

Still further with respect to Claim 13, it is urged that none of the cited references teach or suggest any type of rollback operation, for similar reasons to those given above with respect to Claim 1 and the restoring step. Thus, there is at least one additional missing claimed step not taught or suggested by the cited references.

Therefore, Claim 13 is shown to have been improperly rejected, as a proper prima facie showing of obviousness has not been established due to the above identified missing claimed elements.

A.9. Claims 15, 40 and 61

With respect to Claim 15, Appellants initially show error in the rejection of such claim for reasons given above with respect to Claim 13 (of which Claim 15 depends upon).

Further with respect to Claim 15, Appellants urge that none of the cited references teach or suggest the claimed feature of "*if an intrusion is absent*, determining whether a time threshold has been reached; and if an absence of a reaching of the time threshold is present, repeating the matching step using another set of actions". In rejecting Claim 15, the Examiner cites the teaching of Conklin at Figure 6-8 and associated text and col. 6, lines 60-63 as teaching this claimed feature. Appellants urge that none of these cited passages describe any type of time determination being made *if an intrusion is absent*, and thus none of these cited passages teach or otherwise suggest (i) determining whether a time threshold has been reached, or (ii) any actions resulting from such (missing) determination. Instead, Conklin states that continuous logging occurs until no packets are written for a predetermined period of time – and this logging is *in response to an intrusion being detected* (col. 6, lines 45-63), which is exactly opposite to the claimed responsive event ("if an intrusion is absent") recited in Claim 15. Thus, Claim 15 is further shown to have been erroneously rejected as a proper prima facie case of obviousness has not been established by the Examiner.

A.10. Claims 22 and 23

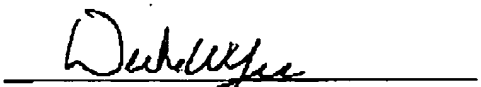
With respect to Claim 22, such claim recites a pattern matcher that "generates a first indication in response to an identification of an absence of an unauthorized intrusion, and generates a second indication to restore the data to a prior state in response to an identification of the unauthorized intrusion". In rejecting Claim 22, the Examiner makes no allegation as to any pattern matcher that performs these two generate functions, and thus a prima facie case of obviousness has not even been alleged by the Examiner, much less established by objective evidence. Thus, Claim 22 is shown to have been erroneously rejected due to such failure to properly establish a prima facie showing of obviousness.

Still further, none of the cited references teach or suggest "a journaler, wherein the journaler journals data in response to accessing of the data and restores the data to the prior state

in response to the indication by the pattern matcher, wherein the data is journaled until the first indication is generated by the pattern matcher". As can be seen, the claimed journaler (i) journals data in response to accessing of the data, and (ii) restores the data to the prior state in response to the indication by the pattern matcher. Thus, a single, unified element (journaler) provides the two explicitly recited operations. The Examiner cites one reference (Conklin) as teaching journaling of data, and recites another reference (Cozza) as teaching data restore. Because two references are relied upon, with each reference allegedly singularly describing a single claimed operation, it necessarily follows that the cited references do not teach or otherwise suggest a single, unitary element (journaler) that itself performs both recited operations. Claim 22 is thus further shown to have been erroneously rejected as there are additional claimed features not taught or suggested by any of the cited references.

Still further, none of the cited references teach the claimed restore of data to a prior state, for similar reasons to those given above with respect to Claim 1 and the cited Cozza reference discussion. Claim 22 is thus further shown to have been erroneously rejected as there are additional claimed features not taught or suggested by any of the cited references.

In conclusion, Appellants have shown that numerous specific claimed features are not taught or otherwise suggested by any of the cited references, and thus urge that Claims 1-67 have been erroneously rejected. Accordingly, Appellants respectfully request that the Board reverse the rejection of Claims 1-67.



Duke W. Yee
Reg. No. 34,285
Wayne P. Bailey
Reg. No. 34,289
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method in a data processing system for protecting data from damage, the method comprising:

journaling the data to form journaled data, wherein journaling the data comprises maintaining a previous state of the data for subsequent, optional restore of the data to the previous state;

determining whether a virus is present in the data processing system after journaling of the data has begun; and

responsive to an identification of the virus, restoring the data using the journaled data.

2. The method of claim 1 further comprising:

responsive to an absence of an identification of the virus, discarding the journaled data.

3. The method of claim 1, wherein the determining step comprises:

performing pattern matching.

4. The method of claim 3, wherein the performing step includes:

comparing a set of actions occurring within the data processing system with a set of patterns.

5. The method of claim 1, wherein the data is located in a storage device external to the data processing system.

6. The method of claim 1 further comprising:

recording a sequence of actions occurring within the data processing system.

7. The method claim 1, wherein the data is data accessed by a process within the data processing system.

8. The method of claim 1 further comprising:

responsive to an identification of the virus, blocking access to the data by a process accessing the data.

9. The method of claim 1 further comprising:

responsive to an identification of the virus, generating an indication halting a process accessing the data.

10. The method of claim 1, wherein the journaled data is accessed by a single process and maintained until a determination is made that the single process is eliminated as a virus candidate.

11. The method of claim 1, wherein the journaled data is stored in a protected memory accessible only by the method.

12. The method of claim 11, wherein the journaled data is stored in a data structure located in a protected memory inaccessible by a process.

13. A method in a data processing system for repairing damage to data, the method comprising:

saving a state of a data object in response to a request to access the data object by a process;

performing pattern matching of a set of actions taken within the data processing system; and

determining whether an unauthorized intrusion has occurred in response to performing pattern matching and if so, initiating a rollback to return the data object back to its saved state.

14. The method of claim 13, wherein the performing step comprises:

comparing the set of actions to a pattern from a set of patterns to form a comparison;

determining whether the comparison indicates that the unauthorized intrusion has occurred; and

responsive to an absence of the unauthorized intrusion, repeating the comparing step using another pattern from the set of patterns.

15. The method of claim 13, wherein the performing step comprises:

matching patterns with the set of actions;

determining whether the unauthorized intrusion has occurred;

if an intrusion is absent, determining whether a time threshold has been reached; and

if an absence of a reaching of the time threshold is present, repeating the matching step using another set of actions.

16. The method of claim 14, wherein match between the pattern and the set of actions identifies an absence of the unauthorized intrusion.

17. The method of claim 14, wherein match between the pattern and the set of actions identifies a presence of the unauthorized intrusion.

18. The method of claim 13, wherein the intrusion is caused by a virus.

19. The method of claim 13, wherein the intrusion is caused by an authorized user input.

20. The method of claim 13 further comprising:

saving a state of all data objects within the data processing system.

21. The method of claim 13, wherein the data object is located in a storage device external to the data processing system.

22. An intrusion protection system for use in a data processing system comprising:

a sensor filter, wherein the sensor filter receives requests to access data within the data processing system from a process;

a pattern matcher, wherein the pattern matcher receives actions initiated by the process, compares the actions to a pattern to form a comparison, determines whether an unauthorized intrusion has occurred, generates a first indication in response to an identification of an absence of an unauthorized intrusion, and generates a second indication to restore the data to a prior state in response to an identification of the unauthorized intrusion; and

a journaler, wherein the journaler journals data in response to accessing of the data and restores the data to the prior state in response to the indication by the pattern matcher, wherein the data is journaled until the first indication is generated by the pattern matcher.

23. The intrusion protection system of claim 22, wherein the intrusion protection system is located within an operating system.

24. A data processing system comprising:

a bus system;

a communications unit connected to the bus system;

a memory connected to the bus system, wherein the memory includes as set of instructions; and

a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to journal the data to form journaled data, wherein journal the data comprises maintaining a previous state of the data for subsequent, optional restore of the data to the previous state; determines whether a virus is present in the data processing system after journaling of the data has begun; and restores the data using the journaled data in response to an identification of the virus.

25. A data processing system comprising:

a bus system;

a communications unit connected to the bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions; and

a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to save a state of a data object in response to a request to access the data object by a process; perform pattern matching of a set of actions taken within the data processing system; and determine whether an unauthorized intrusion has occurred in response to performing pattern matching and if so, initiate a rollback to return the data object back to its saved state.

26. A data processing system for protecting data from damage, the data processing system comprising:

journaling means for journaling the data to form journaled data, wherein the journaling means for journaling the data comprises means for maintaining a previous state of the data for subsequent, optional restore of the data to the previous state;

determining means for determining whether a virus is present in the data processing system after journaling of the data has begun; and

restoring means, responsive to an identification of the virus, for restoring the data using the journaled data.

27. The data processing system of claim 26 further comprising:

discarding means, responsive to an absence of an identification of the virus, for discarding the journaled data.

28. The data processing system of claim 26, wherein the determining means comprises:

means for performing pattern matching.

29. The data processing system of claim 28, wherein the performing means includes:

means for comparing a set of actions occurring within the data processing system with a set of patterns.

30. The data processing system of claim 26, wherein the data is located in a storage device external to the data processing system.

31. The data processing system of claim 26 further comprising:

recording means for recording a sequence of actions occurring within the data processing system.

32. The data processing system claim 26, wherein the data is data accessed by a process within the data processing system.

33. The data processing system of claim 26 further comprising:

blocking means, responsive to an identification of the virus, for blocking access to the data by a process accessing the data.

34. The data processing system of claim 26 further comprising:

generating means, responsive to an identification of the virus, for generating an indication halting a process accessing the data.

35. The data processing system of claim 26, wherein the journaled data is accessed by a single process and maintained until a determination is made that the single process is eliminated as a virus candidate.

36. The data processing system of claim 26, wherein the journaled data is stored in a protected memory accessible only by the method.

37. The data processing system of claim 36, wherein the journaled data is stored in a data structure located in a protected memory inaccessible by the process.

38. A data processing system for repairing damage to data, the data processing system comprising:

saving means for saving a state of a data object in response to a request to access the data object by a process;

performing means for performing pattern matching of a set of actions taken within the data processing system;

determining means for determining whether an unauthorized intrusion has occurred in response to performing pattern matching; and

means for initiating a rollback to return the data object back to its saved state if it is determined that an unauthorized intrusion has occurred.

39. The data processing system of claim 38, wherein the performing means comprises:

means for comparing the set of actions to a pattern from a set of patterns to form a comparison;

means for determining whether the comparison indicates that the unauthorized intrusion has occurred; and

means, responsive to an absence of the unauthorized intrusion, for repeating the comparing step using another pattern from the set of patterns.

40. The data processing system of claim 38, wherein the performing means comprises:

means for matching patterns with the set of actions;

means for determining whether the unauthorized intrusion has occurred;

means, if an intrusion is absent, for determining whether a time threshold has been reached; and

means, if an absence of a reaching of the time threshold is present, for repeating the matching step using another set of actions.

41. The data processing system of claim 39, wherein match between the pattern and the set of actions identifies an absence of the unauthorized intrusion.

42. The data processing system of claim 39, wherein match between the pattern and the set of actions identifies a presence of the unauthorized intrusion.

43. The data processing system of claim 38, wherein the intrusion is caused by a virus.

44. The data processing system of claim 38, wherein the intrusion is caused by an authorized user input.

45. The data processing system of claim 38 further comprising:

saving means for saving a state of all data objects within the data processing system.

46. The data processing system of claim 38, wherein the data object is located in a storage device external to the data processing system.

47. A computer program product in a computer readable medium for protecting data from damage, the computer program product comprising:

first instructions for journaling the data to form journaled data, wherein journaling the data comprises maintaining a previous state of the data for subsequent, optional restore of the data to the previous state;

second instructions for determining whether a virus is present in the data processing system after journaling of the data has begun; and

third instructions, responsive to an identification of the virus, for restoring the data using the journaled data.

48. The computer program product of claim 47 further comprising:

fourth instructions, responsive to an absence of an identification of the virus, for discarding the journaled data.

49. The computer program product of claim 47, wherein the second instructions comprises:

sub-instructions for performing pattern matching.

50. The computer program product of claim 47, wherein the sub-instructions for performing includes:

instructions for comparing a set of actions occurring within the data processing system with a set of patterns.

51. The computer program product of claim 47, wherein the data is located in a storage device external to the data processing system.

52. The computer program product of claim 47 further comprising:

fourth instructions for recording a sequence of actions occurring within the data processing system.

53. The computer program product claim 47, wherein the data is data accessed by a process within the data processing system.

54. The computer program product of claim 47 further comprising:

fourth instructions, responsive to an identification of the virus, for blocking access to the data by a process accessing the data.

55. The computer program product of claim 47 further comprising:

fourth instructions, responsive to an identification of the virus, for generating an indication halting a process accessing the data.

56. The computer program product of claim 47, wherein the journaled data is accessed by a single process and maintained until a determination is made that the single process is eliminated as a virus candidate.

57. The computer program product of claim 47, wherein the journaled data is stored in a protected memory accessible only by the method.

58. The computer program product of claim 57, wherein the journaled data is stored in a data structure located in a protected memory inaccessible by the process.

59. A computer program product in a computer readable medium for repairing damage to data, the computer program product comprising:

first instructions for saving a state of a data object in response to a request to access the data object by a process;

second instructions for performing pattern matching of a set of actions taken within the data processing system;

third instructions for determining whether an unauthorized intrusion has occurred in response to performing pattern matching; and

fourth instructions for initiating a rollback to return the data object back to its saved state if it is determined that an unauthorized intrusion has occurred.

60. The computer program product of claim 59, wherein the second instructions comprises:

first sub-instructions for comparing the set of actions to a pattern from a set of patterns to form a comparison;

second sub-instructions for determining whether the comparison indicates that the unauthorized intrusion has occurred; and

third sub-instructions, responsive to an absence of the unauthorized intrusion, for repeating the comparing step using another pattern from the set of patterns.

61. The computer program product of claim 59, wherein the second instructions comprises:

first sub-instructions for matching patterns with the set of actions;

second sub-instructions for determining whether the unauthorized intrusion has occurred;

third sub-instructions, if an intrusion is absent, for determining whether a time threshold has been reached; and

fourth sub-instructions, if an absence of a reaching of the time threshold is present, for repeating the matching step using another set of actions.

62. The computer program product of claim 60, wherein match between the pattern and the set of actions identifies an absence of the unauthorized intrusion.

63. The computer program product of claim 60, wherein match between the pattern and the set of actions identifies a presence of the unauthorized intrusion.

64. The computer program product of claim 59, wherein the intrusion is caused by a virus.

65. The computer program product of claim 59, wherein the intrusion is caused by an authorized user input.

66. The computer program product of claim 59 further comprising:

fourth instructions for saving a state of all data objects within the data processing system.

67. The computer program product of claim 59, wherein the data object is located in a storage device external to the data processing system.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.